8

به نام خدا

دانشگاه تهران

پردیس دانشکده های فنی

دانشکده مهندسی برق و کامپیوتر

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

تمرین شماره‌ی 1

بخش عملی

**استاد درس:**

دکتر فدائی

دکتر یعقوب‌زاده

**نگارش:**

فاطمه محمدی 810199489

[اهداف: 3](#_Toc162646227)

[مقدمه: 3](#_Toc162646228)

[تعریف پروژه: 3](#_Toc162646229)

[داده‌ها و پارامتر‌های مسئله: 3](#_Toc162646230)

[پیاده سازی گام به گام پروژه: 3](#_Toc162646231)

[بخش صفر: مروری بر ورودی ها، کتابخانه ها و داده های ضروری: 3](#_Toc162646232)

[تعریف کتابخانه‌های مورد نیاز: 3](#_Toc162646233)

[تعریف ورودی‌ها: 3](#_Toc162646234)

[بخش اول: پیش پردازش و استخراج ویژگی: 4](#_Toc162646235)

[پیش پردازش: 4](#_Toc162646236)

[استخراج ویژگی: 4](#_Toc162646237)

[بخش دوم: آشنایی با HMM: 5](#_Toc162646238)

[بخش سوم: پیاده سازی مسئله: 5](#_Toc162646239)

[نتیجه گیری نهایی: 5](#_Toc162646240)

[راهکار برای توسعه و بهبود پروژه: 5](#_Toc162646241)

[سوالات: 5](#_Toc162646242)

[منابع استفاده شده: 5](#_Toc162646243)

پردازش سیگنال

Hidden Markov Models

# اهداف:

توسعه یک سیستم تشخیص گفتار اعداد براساس HMM میباشد.

# مقدمه:

Hidden Markov Models (HMM) یکی از ابزارهای مهم در زمینه پردازش سیگنال های زمانی و تشخیص الگو هستند. در این مدل ها هر حالت به یک وضعیت مشخص مرتبط می شود و احتمال حرکت از یک وضعیت به وضعیت دیگر با توجه به وضعیت فعلی تعیین می شود. از این رو، HMM به عنوان یک ابزار قدتمند برای مدلسازی سیستم های پویا و تصمیم گیری در شرایط عدم قطعیت به شمار می آید. در زمینه تشخیص گفتار، HMM معمولاً برای مدل سازی دقیق تر وضعیت های مختلف گفتار، مانند حروف، کلمات یا فریم های زمانی کوتاه از گفتار، استفاده می شود.

# تعریف پروژه:

همانطور که اشاره شد هدف اصلی این پروژه توسعه یک سیستم تشخیص گفتار اعداد براساس HMM میباشد. برای اینکار مجموعه داده صوتی ای استفاده میشود و باید مراحل پیش پردازش، استخراج ویژگی، پیاده سازی مدل و تحلیل آن در طی پروژه انجام داد.

# داده‌ها و پارامتر‌های مسئله:

در این پروژه که یک پروژه داده محور است، مجموعه داده‌ای مورد نیاز، شامل ضبط های گفتاری از 6 گوینده مختلف است که هر کدام 50 بار ارقام 0 تا 9 را به زبان انگلیسی تلفظ می کنند. نام هر فایل صوتی به صورت {digitLabel}\_{speakerName}\_{index}.wav است که به ترتیب عدد گفته شده، نام گوینده و شماره نمونه را نشان میدهد.

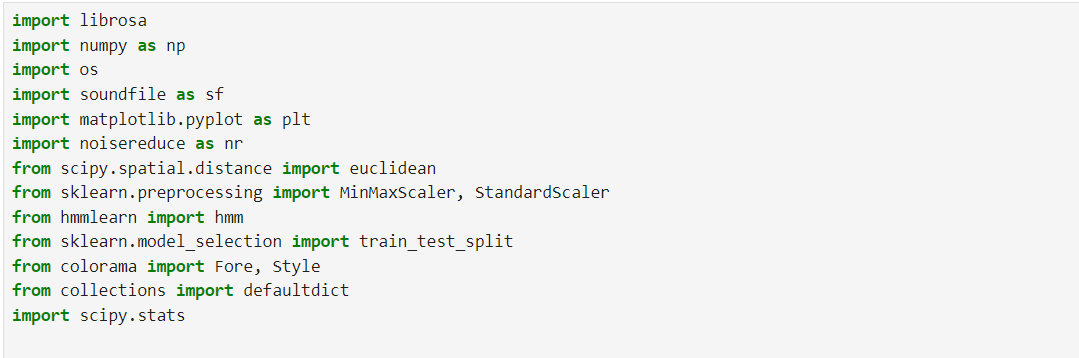
# پیاده سازی گام به گام پروژه:

در ادامه به توضیح هر گام از پروژه به همراه علت انجام آن و تحلیل خروجی توابع صدا زده در هر گام میپردازیم.

## بخش صفر: مروری بر ورودی ها، کتابخانه ها و داده های ضروری:

### تعریف کتابخانه‌های مورد نیاز:

در این پروژه به کتابخانه های زیر نیاز داریم:



در ادامه علت نیاز به هریک از کتابخانه ها را بررسی خواهیم کرد.

### تعریف پارامترها:

در قدم اول پارامترهایی که نیاز داریم را مشخص میکنیم،(برخی از آنها در مراحل جلو تر اضافه میشوند):



* در ابتدا پوشه هایی که نیاز داریم داریم را مشخص میکنیم، که به ترتیب شامل رکورد ها اولیه، رکورد های پیش پردازش شده، MFCCها و Heatmap های MFCC میباشند.
* در ادامه دو set، داریم که مقادیر یکتای از اعداد و یا افرادی که باید تشخیص داده شوند، در آنها ذخیره خواهد شد.
* برای یکنواخت شدن نرخ خواندن رکورد ها و feature MFCC و بهتر بررسی کردن آنها چند معیار باید مشخص کرد، که در ادامه نیز آنها را بیشتر بررسی میکنیم، این معیار ها شامل نرخ نمونه برداری از داده، N\_MFCC و N\_FRAME\_MFCC است که در ادامه به بررسی این دو خواهیم پرداخت.
* در ادامه تعداد رکورد ها به ازای هر رقم که فردگفته را مشخص میکنیم.
* برای تقسیم داده ها به دو دسته Train و Test نیز باید درصد هر کدام را مشخص کنیم.
* در اخر برای مدل های HMM تعداد Hidden State ها و تعداد تکرار (برای Train) را مشخص میکنیم.

## بخش اول: پیش پردازش و استخراج ویژگی:

### پیش پردازش:

پیش پردازش داده ها اولین گام در هر پروژه داده محور است. هدف از پیش پردازش داده ها، بهبود کیفیت داده ها و آماده سازی آنها برای آموزش و آزمایش سیستم تشخیص گفتار است. مراحل معمول پیش پردازش داده ها می تواند شامل افزایش داده، حذف نویز، تقویت سیگنال، نرمال سازی و قطعه بندی باشد.

#### پرسش 1:

آیا قطعه بندی داده ها برای این دیتاست مفید است؟ چرا؟

پاسخ:

* در صورتی که منظور قطعه بندی هر وویس باشد، پاسخ خیر است، چرا که در هر فایل ضبط شده تنها یک رقم توسط یک نفر گفته شده است. درواقع قطعه بندی برای آن است که درصورتی که در یک فایل چندین فرد صحبت کرده باشند و یا چندین رقم بیان شده باشد، آنها را جدا کنیم، که در این دیتاست این کار از قبل انجام شده است.
* درصورتی که منظور دسته بندی وویس ها به چند گروه براساس گوینده یا عدد گفته شده باشد، میتوانیم این دسته بندی را داشته باشیم تا سریع تر داده ها را را بررسی کنیم، البته توجه شود با توجه به این که در این پروژه به دو هدف از داده ها استفاده میکنیم و همچنین داده هایی که برای Train هستند را فقط میتوان دسته بندی کرد، این عمل را در فاز های بعدی انجام میدهیم. (توجه شود در واقعیت مواردی هستند که داده های مورد تست ناشناخته هستند و نمیتوان دسته بندی کرد.)

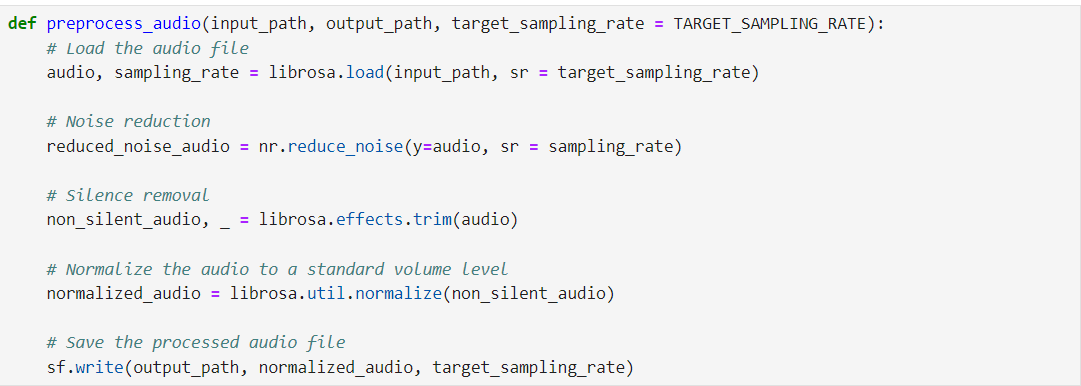
البته این کار را میتوانستیم در این فاز هم انجام بدهیم، اما فعلا آن را بعد از دسته بندی داده ها به دو گروه Train , Test انجام میدهیم.

دلیل آنکه قبل از این فاز دسته بندی به دو گروه Test و Train را انجام نداده ایم ساده تر شدن برنامه است چرا که تمامی مراحل این فاز باید روی همه داده ها (هم Testو هم Train) انجام شوند.

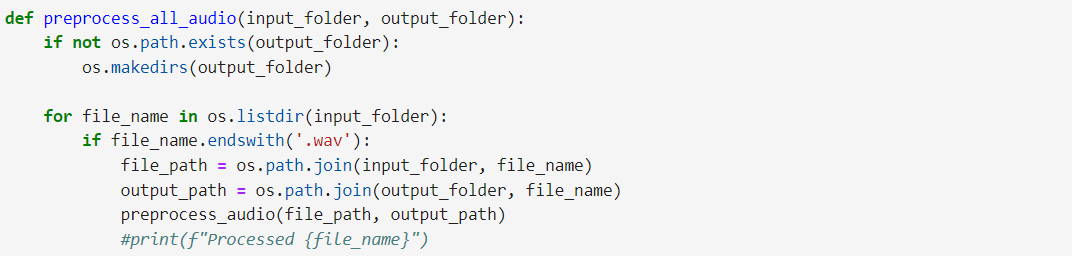
* دلایل اهمیت قطعه بندی:

در این بخش همانطور که اشاره شده ما ابتدا برخی از پیش پردازش ها و استخراج ویژگی ها را انجام میدهیم، اما قطعه بندی را فعلا انجام نمیدهیم:

1. در گام اول یک تابع تعریف میکینم که ابتدا یک تابع تعریف میکنیم که برای یک رکورد پیش پردازشی شامل 1. کاهش نویز، 2. حذف بخش silent 3. نرمال سازی را انجام میدهد:



1. سپس تابعی مینویم که این عمل را برای تمام رکورد های اولیه انجام دهد و آنها را در پوشه دیگری ذخیره کند:



1. در اخرین قدم این تابع را فراخوانی میکنیم:



### استخراج ویژگی:

استخراج و انتخاب ویژگی از مهم ترین مراحل هر پروژه هوش مصنوعی و یادگیری ماشین می باشند. استفاده از ویژگی مناسب می تواند تاثیر بالایی در خروجی مدل نهایی داشته باشد.

همچنین ویژگی های بسیار متفاوتی را می توان از یک محتوای صوتی استخراج کرد که هر کدام اطلاعات گوناگونی را به ما می دهند. از جمله این ویژگی ها می توان mel-spectrogram، Zero Crossing Rate، mfcc و chroma features را نام برد.

#### پرسش 2:

در مورد هر کدام از این ویژگی ها تحقیق کنید و روابط بین آن ها را توضیح دهید.

پاسخ:

* همانطور که قبلتر اشاره کرده ایم، استخراج ویژگی یک مرحله مهم در پردازش صدا و وظایف تشخیص گفتار است که داده‌های صوتی خام را به نمایشی آموزنده‌تر و فشرده‌تر تبدیل می‌کند. که از جمله این ویژگی ها می توان mel-spectrogram، Zero Crossing Rate، mfcc و chroma features را نام برد.

در ادامه به توضیح مختصری درباره ی هریک از این ویژگی ها میپردازیم:

**1. Mel-Spectrogram**

mel-spectrogram یک نمایش بصری از طیف فرکانس‌های یک سیگنال صوتی است که با زمان تغییر می‌کنند و یک مقیاس mel در محور فرکانس دارد. مقیاس mel به گونه ای طراحی شده است که پاسخ گوش انسان را بیشتر از باندهای فرکانسی با فاصله خطی مورد استفاده در طیف سنجی استاندارد تقلید کند. این باعث می‌شود که طیف‌نگار mel به ویژه برای پردازش گفتار و موسیقی مفید باشد زیرا روی فرکانس‌هایی که برای درک شنوایی انسان مهم هستند تأکید می‌کند..

**2. Zero Crossing Rate (ZCR)**

ZCR اندازه گیری تعداد دفعاتی است که سیگنال صوتی از محور صفر در یک بازه زمانی معین عبور می کند. این یک ویژگی ساده و در عین حال موثر برای تجزیه و تحلیل ویژگی های زمانی صدا است. مقادیر بالای ZCR معمولاً اجزای پر سر و صدا یا با فرکانس بالا را نشان می دهد، در حالی که مقادیر پایین تر نشان دهنده صداهای تونال یا هارمونیک هستند. ZCR می‌تواند در تمایز بخش‌های گفتاری صدادار از بی‌صدا، در میان سایر برنامه‌ها مفید باشد.

**3. Mel-Frequency Cepstral Coefficients (MFCCs)**

MFCC ها ضرایبی هستند که در مجموع یک MFC را تشکیل می دهند. آنها از نمایش واقعی مغزی کلیپ صوتی مشتق شده اند، که در آن باندهای فرکانس خطی نیستند اما بر اساس مقیاس mel توزیع می شوند. این تقریباً پاسخ سیستم شنوایی انسان را دقیق‌تر از هر طیف‌نگار فرکانس خطی تقریب می‌کند. MFCC به طور گسترده ای در تشخیص گفتار و طبقه بندی ژانر موسیقی استفاده می شود، زیرا آنها به طور موثر ویژگی های صدا و متن سیگنال های صوتی را ضبط می کنند.

**4. Chroma Features**

ویژگی‌های Chroma ابزاری قدرتمند برای تجزیه و تحلیل موسیقی است که شدت هر یک از ۱۲ کلاس مختلف را بدون توجه به اکتاو(octave) نشان می‌دهد. برای تشخیص گفتار، ویژگی‌های کروما کمتر به طور مستقیم مورد استفاده قرار می‌گیرند، اما می‌توانند بینش‌هایی را در مورد میزان آهنگ و محتوای هارمونیک در سیگنال‌های صوتی که ملودی و هارمونی مرتبط هستند، ارائه دهند.

**روابط و کاربردها**

* + **Mel-Spectrogram و MFCCs:**

هر دو بر اساس مقیاس mel هستند که برای بازتاب شنوایی انسان طراحی شده است.

در حالی که mel-spectrogram ها نمایش متراکمی از سیگنال مفید برای تحلیل بصری و مدل های یادگیری عمیق را ارائه می دهند، MFCC ها این اطلاعات را در مجموعه فشرده ای از ویژگی ها خلاصه می کنند که مهم ترین جنبه های صدا را همانطور که توسط انسان درک می شود، به تصویر می کشد، و آنها را برای مدلهای یادگیری ماشین سنتی بسیار کارآمد می کند.

* + **ZCR and Chroma Features:**

ZCR اندازه گیری ساده ای از تغییرات محتوای فرکانس سیگنال را ارائه می دهد که برای شناسایی بخش های گفتار یا تشخیص ریتم در موسیقی مفید است. Chroma Features ، اگرچه پیچیده‌تر هستند، اما نمای دقیق‌تری از محتوای هارمونیک ارائه می‌دهند. هر دو می توانند مکمل یکدیگر در کاربردهایی باشند که از درک هر دو جنبه ریتمیک و هارمونیک صدا، مانند سیستم های آنالیز موسیقی پیشرفته بهره می برند.

**تشخیص گفتار:**

برای تشخیص گفتار، MFCCها به دلیل اثربخشی آنها در گرفتن ویژگی های کلیدی گفتار، اغلب انتخاب اصلی هستند. می توان از Mel-spectrograms نیز استفاده کرد، به خصوص با deep learning models که می توانند از نمایش متراکم آنها یاد بگیرند. ZCR می‌تواند به تقسیم‌بندی گفتار از سکوت یا نویز کمک کند، در حالی که ویژگی‌های chroma کمتر در برنامه‌های گفتاری خالص استفاده می‌شوند، اما می‌توانند زمینه اضافی را در وظایف مربوط به آواز خواندن یا جایی که اطلاعات زیر و بم مرتبط هستند، فراهم کند.

به طور خلاصه، هر یک از این ویژگی ها نقاط قوت و ضعف خود را دارند و برای جنبه های خاصی از تجزیه و تحلیل صدا مناسب هستند. MFCCها و mel-spectrograms ارتباط نزدیکی دارند و برای تشخیص گفتار بسیار مرتبط هستند، در حالی که ZCR در تشخیص و تقسیم‌بندی گفتار ارزش می‌افزاید. ویژگی‌های Chroma، در حالی که در زمینه گفتار جا افتاده‌تر هستند، بینش‌هایی را در مورد محتوای هارمونیک ارائه می‌دهند که می‌توانند سایر ویژگی‌ها را در برنامه‌های دارای تمایل به موسیقی تکمیل کنند.

### استخراج MFCC:

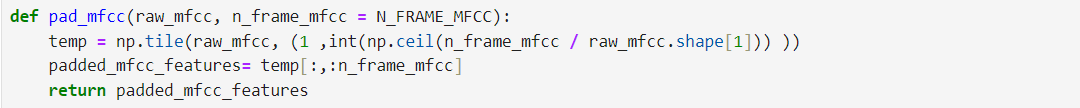
در این پروژه از ضرایب Mel Frequency Cepstral Coefficient (MFCC) برای دسته بندی record ها استفاده میکنیم.

ضرایب MFCC به طور گسترده در زمینه های مرتبط با دسته بندی موسیقی و تشخیص گفتار استفاده شده اند.

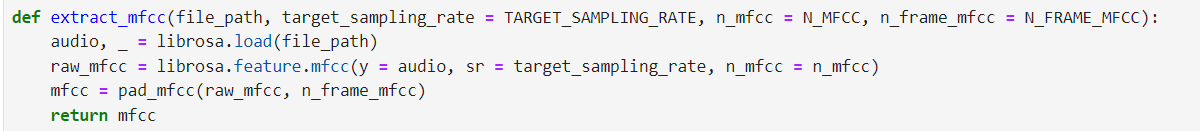
در ادامه مجموعه ضرایب MFCC را برای هر کدام از نمونه های داده شده استخراج میکنیم.

1. یکی از اقدام مهم برای داشتن MFCCs هایی با shape های یکسان، حفظ ساختار زمانی است که میتوان به صورت های متفاوت انجام شود که در اینجا ما از روش padding استفاده میکنیم.

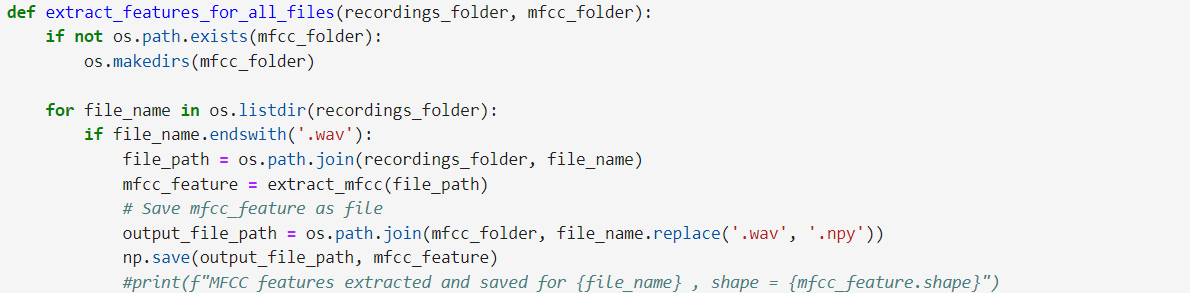
padding نه فقط برای دستیابی به طول ثابت، بلکه برای حفظ ساختار زمانی سیگنال صوتی در آرایه ویژگی استفاده می شود. در اینجا ما از N\_FRAME\_MFCC که به صورت تجربی برابر با 30 قرار داده ایم استفاده میکنیم تا تمام MFCC هایی خود را به صورتی یکدست تولید کنیم، به این منظور تابع زیر را تعریف میکنیم:



1. در ادامه تابعی تعریف میکنیم که به ازای هر فایل رکورد، ابتدا با کمک کتابخانه librosa، mfcc فایل را به صورت خام استخراج میکند و سپس به کمک تابع pad\_mfcc آن را به فرمت خواسته شده در می اورد.



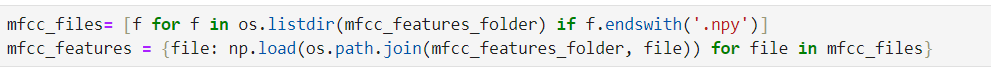
1. در ادامه نیز یک تابع نوشته ایم که با ازای تمامی رکورد های موجود در یک فولدر mfcc ها را استخراج کند و آنها را در فولدر دیگری ذخیره کند.



1. سپس با فراخوانی تابع بالا، mfcc ها را استخراج میکنیم.



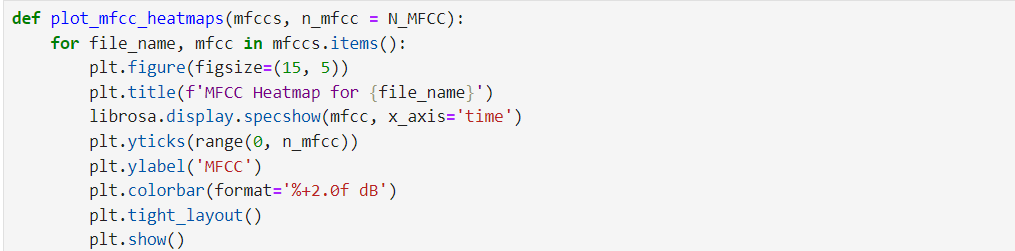
1. میتوانسیتم در همان تابع بالا در زمان ذخیره mfcc آنها در یک set نیز ذخیره کنیم، که در کل سرعت برنامه را شاید بیشتر میکرد، اما به دلیل تک کاربردی بودن هر تابع، این کار را در ادامه به صورت زیر با خواندن مجدد فایل هایی که تولید شده اند انجام میدهیم:



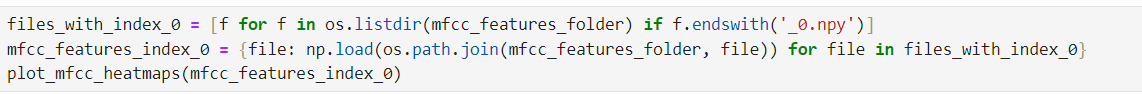
### رسم Heat Map:

پس از استخراج ویژگی ها، نمودار Heat Map مربوط به به ضرایب برای یک نمونه از هرکدام از دسته ها را رسم میکنیم.

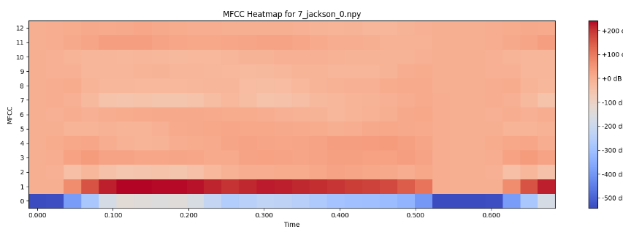
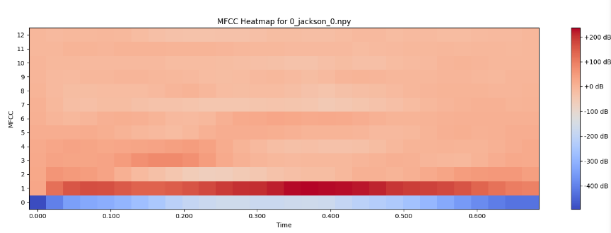
1. به منظور رسم نمودار های Heat Map، تابع زیر را تعریف کرده ایم که به کمک کتابخانه matplotlib میتوان Heat Map را برای تمامی mfcc ها در یک mfcc set رسم کرد:

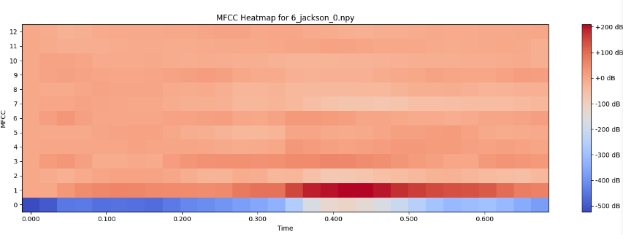
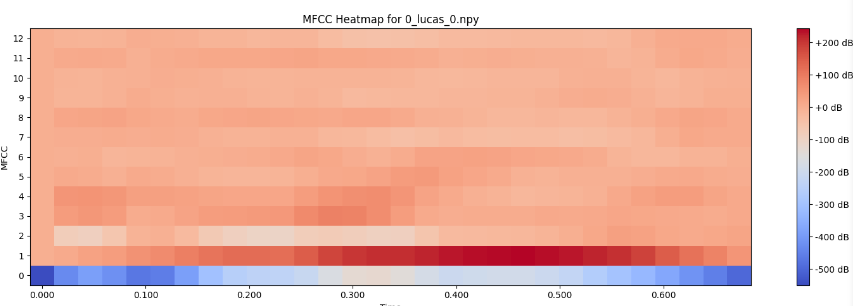


1. در ادامه برای آنکه تنها یک فایل به ازای هر رقم گفته شده توسط هر فرد را رسم کنیم، mfcc ها را براساس index دسته بندی میکنیم و فقط برای آنهایی که index آنها 0 هستند را رسم میکنیم.

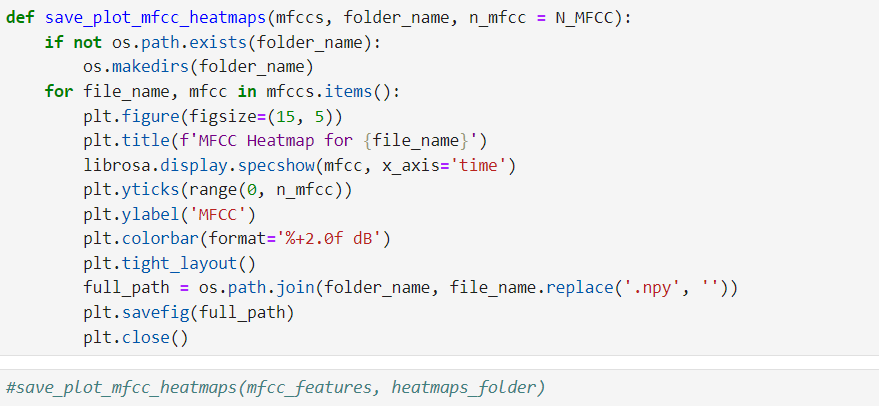


در ادامه برخی از نمودار ها را اورده ایم:





در ادامه برای آنکه بتوان تمامی نمودار ها را دریافت و ذخیره کرد تابع زیر را نوشته ایم، که به دلیل جلوگیری از کند شدن برنامه فعلا به اجرای آن نمیپردازیم:



#### پرسش 3:

Robustness و حساسیت ویژگی های MFCCs را نسبت به تغیرات در سیگنال های صوتی بررسی کنید:

پاسخ:

#### پرسش 4:

آیا موارد خاصی وجود دارند که ضرایب MFCC کارایی کمتری داشته باشند؟

پاسخ:

#### پرسش 5:

چرا در محاسبه MFCC فریم های استفاده شده با یکدیگر هم پوشانی دارند؟

پاسخ:

#### پرسش 6:

چرا در اکثر پروژه های مرتبط با صوت تنها از ۱۲ یا ۱۳ ضریب ابتدایی MFCC استفاده میشود؟

پاسخ:

## بخش دوم: آشنایی با HMM:

#### پرسش 1:

توضیح دهید منظور از State ها و Observation چیست؟

پاسخ:

در این تمرین State ها کدامند و Observation چگونه بدست می‌آید؟

پاسخ:

#### پرسش 2:

مدل های HMM را میتوان بر اساس میزان وابستگی میان State های پنهان دسته بندی کرد، مدلی که در این تمرین به پیاده سازی آن می پردازید یک مدل First-Order HMM است. دلیل نامگذاری آن و همچنین ویژگی های آن را بررسی کنید و تفاوت آن با مدل های دیگر در این دسته بندی را بیان کنید.

پاسخ:

#### پرسش 3:

درباره HMM تحقیق کنید و توضیح دهید که که این مدل برای بررسی و تحلیل چه پدیده هایی مناسب است؟ چرایی این موضوع را توضیح دهید.

پاسخ:

#### پرسش 4:

مدل HMM نیز مانند هر مدل دیگری دارای مزایا و معایبی است که آن را ویژه می کند. مزایا و معایب این مدل را بررسی کرده و هر کدام را مختصرا توضیح دهید.

پاسخ:

#### پرسش 5:

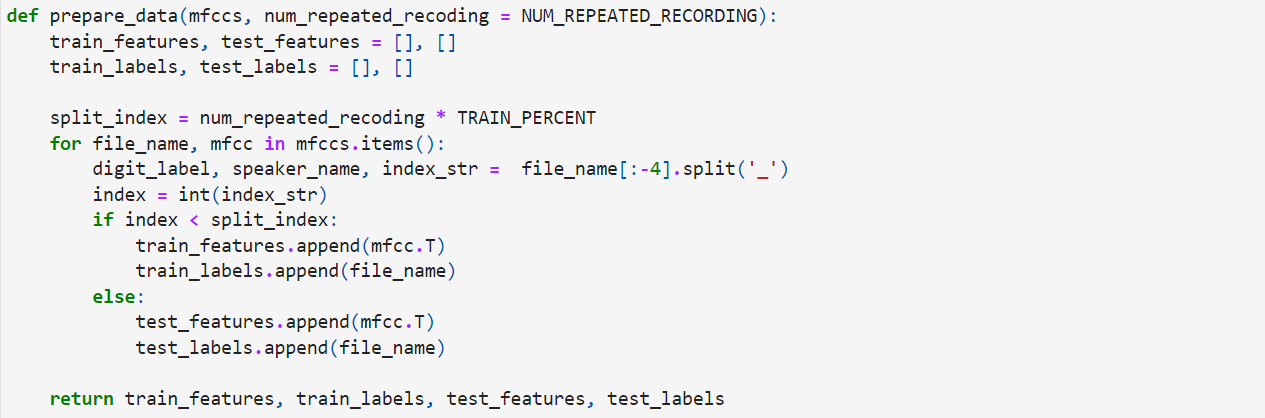
انواع مختلفی از مدل های HMM وجود دارد، درباره آنها تحقیق کنید و چند مورد را بطور مختصر بررسی کنید.

پاسخ:

### اماده سازی داده ها:

در ابتدا برای ورود به فاز modeling و پیاده سازی، لازم است، داد ها را به دو دسته برای یادگیری (Train) و آزمایش (Test) تقسیم کنیم.













## بخش سوم: پیاده سازی مسئله:

### پارت اول: پیاده سازی به کمک کتابخانه ها:

### پارت دوم: پیاده سازی به کمک Scratch:

## بخش چهارم: ارزیابی و تحلیل

معیارهای زیادی برای سنجش و ارزیابی عملکرد مدل ها وجود دارد؛ مثل: Score Accuracy، Precision، Recall، F1

#### پرسش 1:

پاسخ:

#### پرسش 2:

پاسخ:

#### پرسش 3:

پاسخ:

#### پرسش 4:

پاسخ:

#### پرسش 5:

پاسخ:

#### پرسش 6:

پاسخ:

#### پرسش 7:

پاسخ:

#### پرسش8:

پاسخ:

# نتیجه گیری نهایی:

# راهکار برای توسعه و بهبود پروژه:

# سوالات:

# منابع استفاده شده:

YouTube:

* [(ML 14.7) Forward algorithm (part 1) (youtube.com)](https://www.youtube.com/@mathematicalmonk)
* [Hidden Markov Models 09: the forward-backward algorithm (youtube.com)](https://www.youtube.com/@djp3)
* [Forward Algorithm Clearly Explained | Hidden Markov Model | Part - 6 (youtube.com)](https://www.youtube.com/@NormalizedNerd)